Devoir maison - Deep Learning

À rendre au plus tard le 31 mars

I. Batch Normalization

Une couche de batch normalisation prend en entrée la sortie de la couche précédente et la normalise par batch puis la transforme par une fonction affine. Elle est en général suivie d'une fonction d'activation de type Relu. On note $x_i, i = 1..m$ les valeurs d'activation d'un neurone de la couche précédente sur un batch de taille m. Les x_i sont centrés réduits et transformés en $\hat{x}_i$ à partir de statistiques calculées sur un batch B suivant, avec $\epsilon > 0$:

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \qquad \mu_B = \frac{1}{m}\sum_i x_i \qquad \sigma_B = \frac{1}{m}\sum_i (x_i - \mu_B)^2$$

La sortie de la couche de Batch norm est:

$$y_i = \gamma \hat{x}_i + \beta.$$

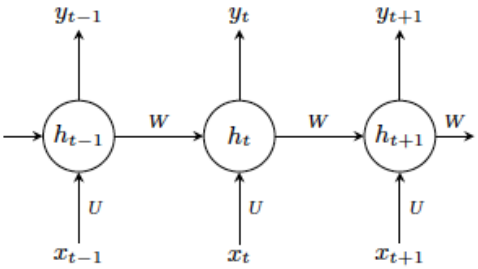
Pour utiliser la bacth normalisation il faut pouvoir apprendre les paramètres de la batch normalisation et propager le gradient à travers cette couche.

(1) On suppose que l'on sait calculer le gradient du critère d'erreur en sortie du réseau, noté l comme *loss*, par rapport aux sorties de la batch normalisation layer $\frac{\partial l}{\partial y_i}$. Calculer le gradient de l par rapport aux paramètres de la batch normalisation γ et β.

(2) Calculer le gradient de l par rapport aux entrées de la batch norm layer $\frac{\partial l}{\partial x_i}$.

II. Propagation du gradient dans les RNN standards

On considère le modèle *RNN* suivant. L'état h_t est calculé suivant : $h_t = \sigma(Wh_{t-1} + Ux_t)$ avec $\sigma(z) = \frac{1}{1+e^{-z}}$.



Soit L la fonction objectif définie comme la somme des L_t à chaque pas de temps: $L = \sum_{t=1}^{T} L_t$, où L_t est un terme de cout réel dépendant de y_t et donc de h_t (voir figure).

(1) Soit $y = \sigma(W \times x)$ avec $y \in \mathbb{R}^n, x \in \mathbb{R}^d, W \in \mathbb{R}^{n \times d}$. Montrez que le Jacobien $\frac{\partial y}{\partial x}$ est égal à $Diag(\sigma') \times W \in \mathbb{R}^{n \times d}$ où vous expliciterez $Diag(\sigma')$.

(2) Montrer que $\frac{\partial L}{\partial W}$ peut s'exprimer comme : $\frac{\partial L}{\partial W} = \sum_{t=0}^{T} \sum_{k=1}^{t} \frac{\partial L_t}{\partial h_t} \frac{\partial h_t}{\partial h_k} \frac{\partial h_k}{\partial W}$.

III. **Evanouissement / explosion du gradient dans les *RNNs*.**

On repart de la seconde question de l'exercice précédent pour étudier sous quelles conditions la norme du gradient peut exploser ou s'évanouir si le nombre de couches augmente.

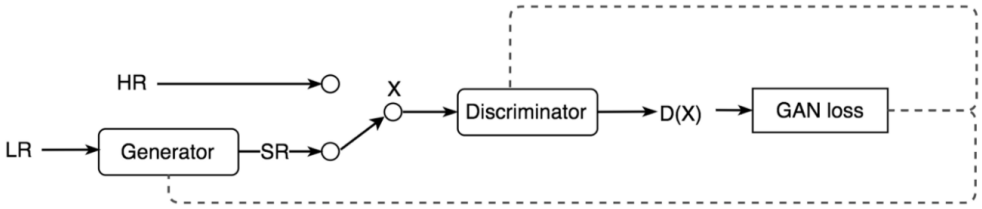
(1) Ré-ecrire $\frac{\partial L}{\partial W}$ en développant dans le cas où $T = 3$. Il devrait apparaître la matrice $[Diag(\sigma') \times W]$ élevée à la puissance. Note : Le diag de σ' n'est pas toujours le meme (abus de notation).

(2) Faire apparaître la matrice $[Diag(\sigma') \times W]$ élevée à la puissance dans le cas général de la seconde question de l'exercice précédent.

(3) On rappelle que toute matrice carrée diagonalisable M peut s'exprimer, par décomposition en vecteurs propres, sous la forme $M = Q\Lambda Q^{-1}$ où Q est une matrice dont la i^{ime} colonne est le i^{ime} vecteur propre de M et Λ est la matrice dans laquelle les valeurs propres correspondantes occupent la diagonale. Montrer que $M^n = Q\Lambda^n Q^{-1}$.

(4) Qu'en déduisez vous sur le calcul du gradient dans un RNN si la longueur de la séquence d'entrée T est grande ? Que se passe-t-il si les valeurs propres sont (en valeur absolue) égales à 1, inférieures à 1, supérieures à 1 ?

IV. **Super Résolution**



(1) La super résolution consiste à générer des données images en haute résolution à partir de données image en basse résolution. Il s'agit d'un problème classique revisité ces dernières années avec le succès des stratégies adversariales.

   a. Quelle stratégie simple non adversariale pourriez vous imaginer pour, par exemple, multiplier par 4 la résolution, c'est à dire doubler le nombre de lignes et de colonnes ?

   b. Quel défaut pouvez vous imaginer (dans les images produites) que votre technique provoquera ? Justifiez.

(2) Le schéma présenté ici contient l'essentiel de l'idée. Il s'agit à partir d'images en Haute et Basse Résolution (HR et BR) d'apprendre une architecture composée d'un générateur (qui calcule une image haute résolution à partir d'une image basse résolution) et d'un discriminateur.

a. Le générateur prend en entrée une image en basse résolution et pas un vecteur latent tiré aléatoirement selon une loi a priori. De quel type de GAN s'agit-il ?

b. En vous inspirant des critères adversariaux déjà étudiés écrivez un critère d'apprentissage pour cette architecture sous forme d'une minimisation de maximisation. Vous préciserez bien quel critère est utilisé pour apprendre chacun des modèles.

c. Cette stratégie montre comment il est possible d'extrapoler une information dont on ne dispose pas initialement. Comment en s'appuyant sur le même principe peut-on concevoir un modèle capable de générer une image 3D à partir d'une image 2D ? De quel genre de jeu de données a-t-on besoin ?